

UNIVERSIDADE EVANGÉLICA DE GOIÁS - UNIEVANGÉLICA
ENGENHARIA DE SOFTWARE

FAGNER OCTAVIO FRANK DO NASCIMENTO
JEFFERSON DIOGO ARAUJO DOS SANTOS

Banco De Dados: A Integração De Características ACID Em Bancos NoSQL

Anápolis
novembro, 2021

UNIVERSIDADE EVANGÉLICA DE GOIÁS - UNIEVANGÉLICA
ENGENHARIA DE SOFTWARE

FAGNER OCTAVIO FRANK DO NASCIMENTO
JEFFERSON DIOGO ARAUJO DOS SANTOS

Banco De Dados: A Integração De Características ACID Em Bancos NoSQL

Trabalho apresentado ao Curso de Engenharia de Software da
Universidade Evangélica de Goiás – UniEVANGÉLICA, da
cidade de Anápolis-GO como requisito parcial para obtenção do
Grau de Bacharel em Engenharia de software.

Orientador (a): Prof. Natasha Sophie Pereira

Anápolis
novembro, 2021

FICHA CATALOGRÁFICA

NASCIMENTO, F. O. F. **Banco De Dados: A Integração De Características ACID Em Bancos NoSQL**. Anápolis, 2021. (Universidade Evangélica de Goiás – UniEVANGÉLICA, Engenheiro(a) de Software, 2021).

Monografia. Universidade Evangélica de Goiás, Curso de Engenharia de Software, da cidade de Anápolis-GO.

SANTOS, J. D. A. **Banco De Dados: A Integração De Características ACID Em Bancos NoSQL**. Anápolis, 2021. (Universidade Evangélica de Goiás – UniEVANGÉLICA, Engenheiro(a) de Software, 2021).

Monografia. Universidade Evangélica de Goiás, Curso de Engenharia de Software, da cidade de Anápolis-GO.

1. NoSQL, ACID, Banco de dados.

REFERÊNCIA BIBLIOGRÁFICA

NASCIMENTO, F. O. F. SANTOS, J. D. A. **Banco De Dados: A Integração De Características ACID Em Bancos NoSQL**. Anápolis, 2021. P 34. Monografia - Curso de Engenharia de Software, da cidade de Anápolis-GO. Universidade Evangélica de Goiás - UniEVANGÉLICA.

CESSÃO DE DIREITOS

NOMES DOS AUTORES: NASCIMENTO, F. O. F. SANTOS, J. D. A.

TÍTULO DO TRABALHO: Banco De Dados: A Integração De Características ACID Em Bancos NoSQL.

GRAU/ANO: Graduação /2021

É concedida à Universidade Evangélica de Goiás - UniEVANGÉLICA, permissão para reproduzir cópias deste trabalho, emprestar ou vender tais cópias para propósitos acadêmicos e científicos. O autor reserva outros direitos de publicação e nenhuma parte deste trabalho pode ser reproduzida sem a autorização por escrito do autor.

Resumo

INTRODUÇÃO: O fato de outros bancos de dados não relacionais começarem a utilizar as características ACID (Atomicidade, Consistência, Isolamento, Durabilidade) em seus bancos, benefícios que antes haviam dispensado, por conta que deveria ser um banco com propostas diferentes dos bancos relacionais. Os bancos de dados NoSQL são bancos de dados não relacionais que apareceram com maior prioridade após o BigData, para suprir as necessidades do momento. Os bancos NoSQL são conhecidos por não utilizarem ACID, devido a isto possuem uma maior velocidade e flexibilidade ao armazenar e visualizar os dados, pergunta-se: Por que alguns SGBDs *NoSQL* implantaram características ACID em seus bancos de dados?

RESULTADOS E DISCUÇÕES: Abordando eventos históricos relacionados ao SGBD O conceito de bancos de dados existia antes da introdução dos computadores em meados do século XX, mas ainda não tinham essa denominação. No final da década de 70 um sistema foi desenvolvido e nomeado como Sistema R. também foi desenvolvida como linguagem padrão para os bancos de dados relacionais conhecida como SQL. ACID é o conjunto de propriedades de transação em banco de dados. Todas as operações que são realizadas em qualquer base de dados relacionais utilizam essas características tornando o banco seguro e confiável. NoSQL contêm diferentes sistemas de armazenamento que vieram para suprir necessidades em demandas ocorridas pelo efeito da Big Data. O termo Big Data é utilizado para descrever a grande quantidade de dados armazenados em servidores vindos de diversas fontes de dados como mídias sociais como Twitter, Facebook, e-mails, sensores, e assim por diante. NoSQL foram pensados para não ter um padrão e especialmente para não ter a linguagem SQL.

MÉTODOLOGIA: A metodologia partirá de estudo bibliográfico, utilizando para isso recursos como livros, artigos científicos, explicando o que é um banco de dados não relacional compreendendo os modelos e funcionamento a adesão que os bancos NoSQL fizeram das características ACID

CONSIDERAÇÕES FINAIS: Dessa forma, conclui-se que ACID nos SGBDs relacionais tem um papel importante, pois essas características ajudam o banco ser integro. Essas características fizeram os bancos NoSQL olharem para elas e começaram a se importar com a consistência do banco, sempre deixar ele valido, adaptando o que a ACID proporciona para o sistema NoSQL.

PALAVRAS-CHAVE: NoSQL, ACID, Banco de dados.

Lista de Imagens

Figura 1 - Modelo Chave-Valor.....	14
Figura 2 - Modelo Família de Colunas.	14
Figura 3 - Modelo de Grafos.....	15
Figura 4 - Modelo de Documento.....	15
Figura 5 - Teorema de CAP.....	17

Lista de Quadros

Quadro 1 - Comparação entre BASE e ACID.....	16
---	----

Lista de Siglas

ACID	Atomicidade, Isolamento, Consistência, Durabilidade;
BASE	<i>Basically Available</i> : Disponibilidade É Prioridade; <i>Soft-State</i> : Não Precisa Ser Consistente O Tempo Todo; <i>Eventually Consistente</i> : Consistente Em Momento Indeterminado;
CAP	<i>Consistency</i> : Consistência; <i>Availability</i> : Disponibilidade; <i>Partition Tolerance</i> : Tolerância Ao Particionamento;
NoSQL	<i>Not Only SQL</i> : Não Apenas SQL;
NewSQL	Nova SQL;
OLTP	<i>On-line Transaction Processing</i> : Processamento De Transações Online
SQL	<i>Structured Query Language</i> : Linguagem de Consulta Estruturada;
SGBD	Sistema Gerenciador de Banco de Dados;

Sumário

INTRODUÇÃO	8
1 FUNDAMENTAÇÃO TEÓRICA	11
1.1 Eventos históricos relacionados ao banco de dados e SGBD	11
1.2 Atomicidade, consistência, isolamento e durabilidade	12
1.3 NoSQL e seus principais modelos	12
1.4 Propriedade BASE	16
1.5 Teorema CAP	16
1.6 BIG DATA	17
1.7 Quando utilizar NoSQL	18
1.8 Adesão do ACID	18
1.9 NewSQL	19
2 METODOLOGIA	20
3 ANÁLISE E DISCUÇÃO DOS RESULTADOS	22
3.1 ARTIGO 1. A Critical Comparison of NOSQL Databases in the Context of ACID and Base.	22
3.2 ARTIGO 2. A middle layer solution to support ACID properties for NoSQL databases.	22
3.3 ARTIGO 3. A proposal of a transaction processing method for MongoDB.	23
3.4 ARTIGO 4. What are ACID Transactions?	23
3.5 ARTIGO 5. SQL vs. NoSQL vs. NewSQL- A Comparative Study.	24
3.6 CATEGORIA 1: Funcionamento CAP X BASE X ACID	25
3.7 CATEGORIA 2: Integração da ACID em Bancos NoSQL	25
4 CONSIDERAÇÕES FINAIS	27
REFERÊNCIAS BIBLIOGRÁFICAS	28

INTRODUÇÃO

PROBLEMA

O Sistema gerenciador de banco de dados (SGBD) relacionais têm intrínseco as propriedades Atomicidade, Consistência, Isolamento, Durabilidade (ACID) que garantem que as transações em banco de dados não tenham duplicatas, onde todas as informações são armazenadas e alteradas simultaneamente por várias pessoas, mas mantendo o isolamento dos dados (PEREIRA, 2015)

Já os SGBDs *Not Only SQL* (NoSQL) apareceram com uma proposta que não utilizaria essas características ACID, sendo mais fácil a implementação em um sistema, mas sem garantir a consistência de dados, porém conseguindo armazenar e gerenciar uma maior quantidade de dados, para tal feito eles utilizam o teorema de CAP (*Consistency, Availability, Partition Tolerance*); e as características BASE (*Basically Available, Soft-State, Eventually Consistente*) (NoSQL, 2009).

Em 2018, o *MongoDB* lançou sua atualização 4.0 contendo características ACID promovendo um diferencial dos demais bancos NoSQL, esse modelo de documento permite que pequenos conjuntos de dados sejam armazenados em um único documento, quando convencionado com a atomicidade de documentos podem extinguir a necessidade de transações projetadas no modelo de dados relacional *MongoDB* (MongoDB,2018). Diante do exposto, o fato de outros bancos de dados não relacionais também utilizaram tais características em seus bancos, assim voltando a utilizar o ACID, benefícios que antes haviam dispensado, pergunta-se:

Porque alguns SGBDs *NoSQL* implantaram características ACID em seus bancos de dados?

OBJETIVO GERAL

As características ACID e uma particularidade dos SGBDs relacionais, os bancos NoSQL surgiram com uma ideia de serem diferentes dos SGBDs relacionais, com o tempo foram adicionando as características ACID em seus sistemas que era uma particularidade dos SGBDs Relacionais.

OBJETIVOS ESPECÍFICOS

- Compreender o modelo dos bancos de dados NoSQL;
- Explicar o que são as características BASE e o teorema de CAP;
- Explicar o que são as características ACID e suas finalidades;
- Avaliar as razões dos bancos NoSQL aderirem às características ACID;

JUSTIFICATIVA

Os SGBDs relacionais são os mais usados tanto nas pequenas empresas como nas multinacionais há cerca de 50 anos, possuem várias ferramentas, sendo muito seguros e respondem às regras funcionais. Mas há apenas 20 anos um novo modelo surgiu sem o uso da linguagem SQL (EQUIPO DE EXPERTOS, 2016).

A não utilização da linguagem SQL conduziu a economia do tempo de tratamento e então o custo financeiro diminuiu, mas é necessário questionar quais os custos e a real necessidade de implementar dentro da aplicação, mesmo a tecnologia que já estava dentro de um SGBD ACID e como assegurar que as diferentes vistas dos dados são iguais, disponíveis e recuperáveis em caso de erro de sistema (SILVA, 2017; THIBAUT, 2015)

Quando os bancos de dados NoSQL surgiram trouxe consigo uma proposta diferente, de ser mais fácil e com menos processos de implementação, para isso utilizam a propriedade BASE e o teorema de CAP. Na propriedade BASE se fizer uma alteração de um valor k no dado x e depois fazer uma consulta nesse mesmo dado x o valor retornado será o valor antecessor ao k , retornando um dado antigo no banco dependendo do tempo da alteração e da consulta. No teorema de CAP dos três pilares existentes somente pode ter a garantia de implementar até dois desses pilares, sendo preciso escolher entre consistência forte, alta disponibilidade e tolerância ao particionamento. As informações que estão no banco que mantenham as características de um banco NoSQL, agora estivesse mais parecido com um banco relacional tendo as características ACID (CIFERRI, 2020).

Os bancos NoSQL propuseram observar que consultas em tempo real, diminuição de falhas, são indispensáveis para qualquer sistema. Considerando que a proposta dos bancos não relacionais era ser diferente, revolucionando o mercado, com a proposta diferente de um banco relacional (GUTIERRY; DEVMEDIA, 2015).

Esse estudo é relevante para explicar o que é um banco de dados não relacional desde o seu surgimento compreendendo os modelos e funcionamento desse tipo de banco como o teorema CAP e características BASE que são as duas propostas que podem ser implementadas nos bancos NoSQL, e posteriormente a adesão que os bancos NoSQL fizeram das características ACID, para garantir um melhor desempenho e consistência nos dados que eles armazenam.

1 FUNDAMENTAÇÃO TEÓRICA

1.1 Eventos históricos relacionados ao banco de dados e SGBD

A ideia de dado é referente ao registro, índice suscetível de análise e interpretação. Os dados são utilizados para a resolução de tarefas e criação de ideias sobre um determinado assunto. O banco de dados está relacionado a um determinado assunto, organizado de maneira útil para que o usuário o utilize retomando informações e tomando decisões claras sobre o assunto (SIMON,1999).

O conceito de bancos de dados existia antes da introdução dos computadores em meados do século XX, mas ainda não tinham essa denominação. Para armazenar as informações, era utilizado arquivos de papel, agrupados em caixas denominadas arquivos. Inicialmente, os registros eram ordenados manualmente. Com a introdução das perfurações, a classificação tornou-se mecânica e depois eletromecânica. O desenvolvimento de bancos de dados gerenciados por meios de computador tornou essas técnicas obsoletas (CODD, 1970)

O termo banco de dados surgiu em meados do ano de 1960 quando as empresas começaram a utilizar computadores como parte efetiva dos custos das empresas. Pois com a chegada dessa tecnologia de forma mais acessível as empresas conseguiram aumentar a capacidade de armazenamento de informações. Nessa época haviam dois modelos de dados: os modelos em rede, denominado *Committee for Data Systems Language* (CODASYL) e o modelo hierárquico *Information Management System* (IMS). Em 1970, Edgar Frank Codd apresentou um modelo de dados relacionais que se tornou um marco, tornando-o referência em banco de dados ao retirar a estrutura lógica do banco de dados do método de armazenamento físico, padronizando o sistema. O avanço significativo em direção ao objetivo da independência de dados foi desenvolvido de forma recente através do fornecimento de tabelas de descrição de dados em um sistema de informação (CODD, 1970).

No final da década de 70 um sistema foi desenvolvido e nomeado como Sistema R. Com a criação desse sistema uma linguagem de consulta também foi desenvolvida como linguagem padrão para os bancos de dados relacionais até os dias atuais, essa linguagem é conhecida como *Structured Query Language* (SQL) (ELMASRI,2004)

Sistema de Gerenciamento de Banco de Dados (SGBD) é o conjunto de softwares responsáveis pelo gerenciamento de um banco de dados que utiliza ferramentas para acessar os

dados que estão armazenado nesse sistema. Os conjuntos de dados são denominados como banco de dados e um dos seus objetivos é promover uma maneira de criar, armazenar, modificar e recuperar de maneira pertinente e eficaz, sendo responsável pela gestão da base de dados (SILBERSCHATZ; KORTH; SUDARSHAN, 2020).

1.2 Atomicidade, consistência, isolamento e durabilidade

É o conjunto de propriedades de transação em banco de dados. Todas as operações que são realizadas em qualquer base de dados relacionais utilizam essas características tornando o banco seguro e confiável (REIS, 2018).

A atomicidade diz respeito a como tudo está sujeito a falhas e caso isso aconteça o sistema deve retornar para o último estado consistente que existia antes da falha. Ao realizar uma determinada operação que vai do ponto A ao B, ao ocorrer falha é possível que ela saia do ponto A, mas não chegue ao ponto B, gerando um estado de inconsistência. A operação tem que acontecer de um ponto ao outro, ou não ocorrer. Consistência é a propriedade que assegura que o banco de dados saía de um estado válido para outro estado válido. Os dados devem ficar com um estado válido para armazenamento de acordo com a regra, por exemplo, ao fazer um registro da venda de um produto, se este não se encontrar cadastrado na lista de produtos do sistema, irá causar uma falha nessa transação. (REIS, 2018).

Já o isolamento trata que todo banco de dados tem operações ocorrendo simultaneamente, informações sendo inseridas, consultadas com o uso dessa propriedade. O banco, por sua vez, deve deixar as informações do mesmo estado caso esses acessos sejam sequenciados, ou seja, se dois clientes estão tentando comprar um único produto disponível em estoque, o primeiro que finalizar a compra vai automaticamente através do sistema interromper a solicitação do outro, esse fenômeno é denominado *rollback*. Por fim, a durabilidade vai garantir que após a execução de uma transação permanecerá neste estado, mesmo que haja problemas no sistema, como falhas e panes elétricas as informações estarão armazenadas em memórias permanentes (Disco Rígido) que ficam disponíveis para uso em intercorrências. (REIS, 2018).

1.3 NoSQL e seus principais modelos

O termo NoSQL apareceu pela primeira vez em 1998, como o nome de um banco de dados relacional de código aberto que não possuía uma interface SQL. NoSQL é diferente do

modelo relacional, por isso deveria ser nomeado NoREL, ou outro termo de mesmo sentido (STROZZI, 1998)

Mas o termo só voltou a ser assunto em 11 de junho de 2009, quando Johan Oskarsson, engenheiro de computação organizou uma reunião em São Francisco para discutir sobre os sistemas de código aberto, distribuídos e não relacionais. Johan Oskarsson queria um nome contundente e fácil de lembrar para esta conferência, foi assim que veio o termo NoSQL (FOWLER, 2012).

NoSQL contém diferentes sistemas de armazenamento que vieram para suprir necessidades em demandas ocorridas pelo efeito da Big Data. Com a chegada de uma enorme quantidade de dados vindo de várias fontes diferentes precisando de tecnologias e técnicas específicas para poder armazenar, analisar e visualizar tais dados, onde os bancos de dados tradicionais são ineficazes. O NoSQL é um banco de dados que foi desenvolvido para armazenar, analisar e visualizar um grande volume de dados auxiliando na resolução de problemas de grandes empresas como Google, Facebook entre outras quando o consumo da internet aumentou drasticamente entre 2012 e 2015 (STROZZI,1998).

A principal característica acerca dos bancos de dados não relacionais é o fato de serem uma solução alternativa para os bancos de dados relacionais como alta escalabilidade e desempenho. A estrutura pouco flexível utilizada até então passou a ser um problema a ser contornado e as soluções propostas tinham como base a eliminação ou minimização dessa estruturação. (BRITOS, 2010). Os Bancos de Dados NoSQL são subdivididos pelo seu núcleo, ou seja, a maneira de como consegue trabalhar com os dados como armazenamento e organização. Dessa forma, os modelos NoSQL são as chave-valor (Figura 1) que por sua vez é modelo que armazena objetos organizados por chaves, esse tipo de banco de dados também é conhecido como tabelas de *hash* distribuídas. Todos os dados possuem uma chave de identificação e seu determinado valor, permitindo assim buscá-los por suas chaves. Este modelo suporta uma grande carga de dados, sendo assim possuem maior escalabilidade. Alguns bancos de dados baseados em chave-valor são: *Table Storage* e *DynamoDB* (SILVA, 2017).

Figura 1: Modelo Chave-Valor

Chave	Valor
carro_3345_cor	preto
carro_3345_pneu	17
carro_3365_cor	branco
carro_3365_pneu	15
carro_4560_peso	1215
carro_4715_ano	2016

Fonte: micreiros.com

Já as famílias de colunas (Figura 2) trata-se de um modelo que armazena os dados em forma de colunas, inspirado pelo *BigTable* do *Google*, eles suportam várias linhas e colunas e cada coluna pode ter subcolunas que possuem várias propriedades. Como na imagem abaixo uma coluna chamada Pessoa_Name, ela vai ter uma coluna chamada ID que é a identificação de cada linha, esse número na coluna ID tem que ser único para cada propriedade, também temos dentro da coluna Pessoa_name a coluna Valor que é preenchida com os respectivos nomes. Além do *BigTable*, outros bancos que usam essa tecnologia são *Cassandra* e *Hypertable* (SILVA, 2017)

Figura 2: Modelo Família de Colunas.

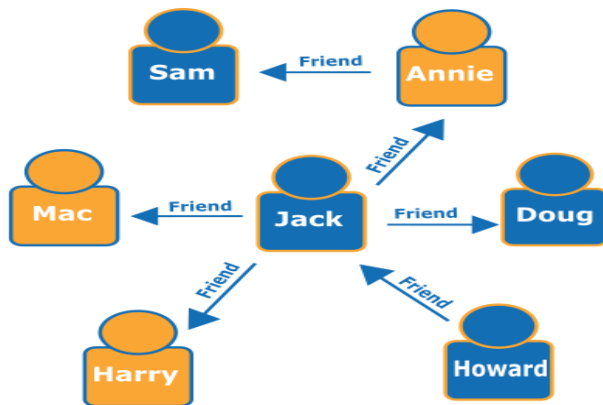
people_id		people_name		people_age	
id	value	id	value	id	value
0	101	0	Mary	0	54
1	102	1	Jhon	1	35
2	103	2	Paul	2	22

Fonte: BARROSO, 2012.

Os grafos (Figura 3) são modelos compostos por vértices e arestas onde os vértices representam as relações. E pode haver relações somente entre pai e filhos, mas há descrição de associação de irmãos, neste caso, o banco de dados pode ser visto como um multi grafo rotulado

e direcionado, ou seja, guardam objetos e não registros como os outros tipos de NoSQL. Alguns bancos de dados que utilizam este conceito são: Neo4j e InfoGrid (SILVA,2017)

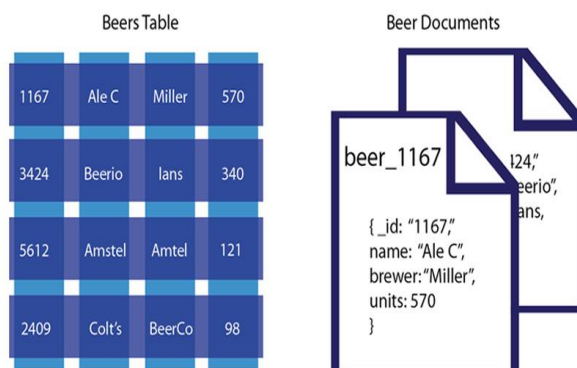
Figura 3: Modelo de Grafos.



Fonte: AWS.

Os modelos documentos (Figura 4) são considerados o que armazena seus registros em formato de documentos JSON ou XML, que podem ser localizados pelo seu ID único ou por qualquer registro que tenha no documento. Portanto, os documentos dos bancos de dados dessa categoria, são conjuntos de atributos e valores, onde um atributo pode ser multivalorado. Em geral, os bancos de dados orientados a documentos não possuem esquema, ou seja, os documentos armazenados não precisam possuir estrutura em comum. Essa característica faz com que seja uma boa opção para o armazenamento de dados semiestruturados. Alguns bancos de dados que utilizam este recurso são MongoDB e RavenDB (ANDERSON, 2009)

Figura 4: Modelo de Documento.



Fonte: Couchbase

1.4 Propriedade BASE

Os bancos de dados NoSQL utilizam a propriedade BASE onde cada letra tem seu significado, BA (*Basically Available*) Disponibilidade é Prioridade. S (*Soft-State*) Não precisa ser consistente o tempo todo. E (*Eventually Consistent*) Consistente em momento indeterminado. Devido à imensa quantidade de dados que são geradas e devem ser analisadas, geradas e entregue a cada requisição, para retirar certas restrições promovidas pelo ACID, como a sincronização de réplicas, para promover a eficiência, se apoiando assim no teorema CAP. Foi feito um comparativo entre BASE e ACID e o foco do BASE é disponibilidade e desempenho como mostrado no quadro abaixo. (VALE; SUDOERS, 2015).

Quadro 1: Comparação entre BASE e ACID.

ACID	BASE
Consistência forte	Consistência fraca
Isolamento	Foco em disponibilidade
Concentra-se em “commit”	Melhor esforço
Transações aninhadas	Respostas aproximadas
Disponibilidade	Mais simples e mais rápido
Conservador (pessimista)	Agressivo (otimista)
Evolução difícil	Evolução fácil

Fonte: VALE; SUDORES,2015.

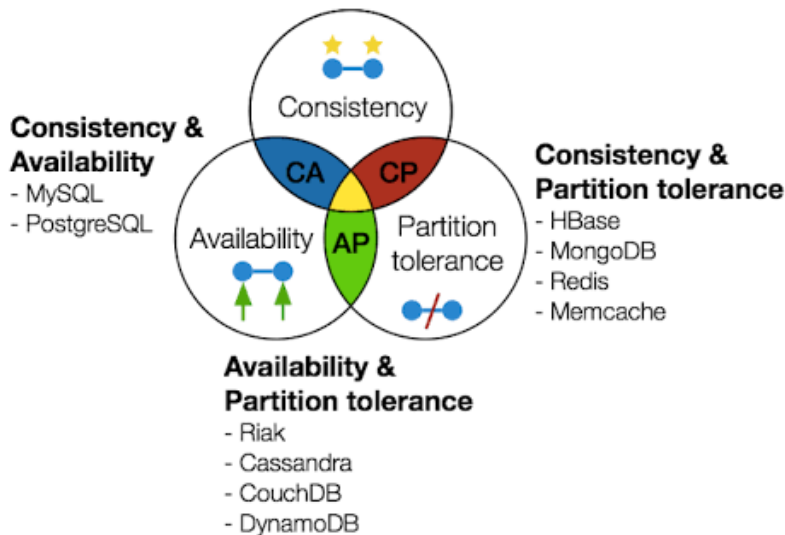
1.5 Teorema CAP

Consistência, Disponibilidade e Tolerância à Partições (CAP) é um teorema que foi criado pelo Dr. Eric Brewer, que define que um sistema de armazenamento de dados não consegue garantir mais de duas das propriedades. É necessário que haja pelo menos duas dessas propriedades para implementar qualquer sistema de dados compartilhados (BREWER, 2000).

É possível que haja utilização simultânea de até duas propriedades (Figura 5) deixando à escolha dos requisitos necessários para determinado sistema. As propriedades são: Consistência: Nesta propriedade é garantido que o usuário irá sempre visualizar os registros no estado mais recente do sistema. Disponibilidade: O sistema está sempre pronto para uso, sem enviar mensagens de erros ou falhas ao usuário do sistema, isto é, Todos os dados ficam

disponíveis para leitura e escrita, mesmo se processos ou ações estiverem em andamento. Tolerância à Partições: O sistema continuará funcionando caso tenha alguma falha ou erro em seus componentes (BREWER, 2000).

Figura 5: Teorema de CAP.



Fonte: SANTOS, 2017.

1.6 BIG DATA

O termo Big Data é utilizado para descrever a grande quantidade de dados armazenados em servidores vindos de diversas fontes de dados como mídias sociais como Twitter, Facebook, e-mails, sensores, e assim por diante. Estes dados que antes não eram utilizados, em 2012 virou extremamente relevante e com isso necessitou a busca por programações que consigam tirar grande proveito dos mesmos, chegando como solução o NoSQL. Banco de dados NoSQL possui mais de uma categoria atendendo as grandes variedades de tipos de dados existentes solucionando tanto as estruturais como as não estruturais (ALECRIM, 2015).

Para a compreensão acerca do termo Big Data é importante entender a definição dos 5Vs: Volume, Variedade, Velocidade, Veracidade, Valor. O Volume significa a grande quantidade de dados gerados por sistemas corporativos, por mídias sociais, sensores e outros dispositivos. A variedade representa os dados estruturados e não estruturados, obtidos do Twitter, Facebook, dentre outros. Velocidade representa a resposta quase que em tempo real para agir no próprio evento gerador das informações. A veracidade determina a relevância dos dados disponíveis, de forma que essas informações possam servir de guia para realização do

planejamento com uma maior segurança. Valor onde é definido a abordagem utilizada para coleta dos dados que estão circulando (TAURION *apud* KOO,2014)

Os benefícios do uso do Big Data podem ser vistos, por exemplo, em uma empresa de vendas. O qual possui o armazenamento de grande quantidade de dados que o Big Data possibilita, é possível ter o perfil de cada cliente, sabendo o produto mais procurado, época onde houve mais vendas, e demais informações, utilizando as características de cada um no momento da venda de seu produto. O termo Big Data vem chamando atenção pela acelerada escala em que os volumes de dados criados pela sociedade a cada dia que passa vêm se tornando cada vez maior. Através do Big Data é possível a utilização de diversas tecnologias de gerenciamento, para administrar essa grande escala de dados (TAURION *apud* KOO,2014)

1.7 Quando utilizar NoSQL

Os bancos de dados NoSQL surgiram para suprir uma crescente demanda de dados que vinha decorrente do BigData. No banco NoSQL conseguimos construir bancos de documentos com diversos atributos sem a necessidade de realizar várias tabelas e juntá-las para assim realizar tudo em somente um documento. Em um certo momento os bancos relacionais ficam grandes devido aos grandes níveis de informações, isso pode acarretar demora nas buscas dos dados, usando os NoSQL o tempo de gravar e recuperar dados é menor comparado aos modelos relacionais (VALE; SUDOERS, 2015).

1.8 Adesão do ACID

NoSQL foram pensados para não ter um padrão e especialmente para não ter a linguagem SQL, como esse banco se tornou muito popular diversos problemas começaram a surgir, não tinha uso de transações, os dados tinham duplicatas, não tinha linguagem de consulta estruturada, e sem ter a modelagem de dados eles começaram a olhar as características dos bancos relacionais (GUTIERRY; DEVMEDIA, 2015).

Os bancos NoSQL que decidiram aderir a essas características ACID sendo chamados de NewSQL, Mike Stonebreaker que é fundador do banco VoltDB, destaca a vantagem dos NewSQL por permitir consulta em tempo real e maior processamento de dados e destacou que é um desperdício não usar a linguagem SQL. A essência dos NoSQL ficou o que mudou foi a

adesão das características ACID tornando o banco mais conciso tendo o conceito de um banco NoSQL, mas com as propriedades de um banco relacional (GUTIERRY; DEVMEDIA, 2015).

1.9 NewSQL

Os bancos de dados NewSQL visam proporcionar os mesmos progressos no desempenho e escalabilidade dos sistemas NoSQL, sem haver prejuízo nos benefícios dos bancos de dados tradicionais, da linguagem SQL e de suas propriedades ACID. NewSQL se define como uma classe de SGBDs relacionais modernos fornecendo o mesmo desempenho de escala dos NoSQL para cargas de trabalho e leitura OLTP (*On-line Transaction Processing*) mantendo as características ACID para as transações (ASLLET; PAVLO, 2016).

Diferentemente dos SGBD tradicionais, que eram percebidos como chave para qualquer tipo de aplicação, os NewSQL utilizam um recurso distinto, através do qual busca-se suprir a partir de cada novo sistema desenvolvido, suprir a uma necessidade específica do mercado, visando atingi-la de maneira separada, encerrando com a antiga ideia de ter um único sistema que sirva para qualquer tipo de aplicação, sem haver a necessidade da criação de um grande volume de funções e comportamentos desnecessários para uma aplicação específica. Há cinco características de um SGBD NewSQL: (STONEBRAKER; CATTEL, 2011, citado por LEMOS; FIGUEIREDO 2014).

- Linguagem SQL como forma de interação entre o SGBD e a aplicação;
- Suporte para transações ACID;
- Controle de concorrência não bloqueante, para que as leituras e escritas não causem conflitos entre si;
- Arquitetura que ofereça um maior desempenho por nó de processamento;
- Arquitetura escalável, com memória distribuída e com capacidade de funcionar em um aglomerado com um grande número de nós.

2 METODOLOGIA

A metodologia partirá de estudo bibliográfico, para compor a pesquisa bibliográfica foram utilizados livros e artigos publicados sobre a temática. Os descritores utilizados na busca dos periódicos foram: ACID; NoSQL; Banco de Dados. Foram selecionados artigos publicados entre 2012 e 2021, cujo idioma era português, inglês e espanhol. Foram excluídos da pesquisa artigos anteriores ao ano de 2012 e em idiomas diferentes dos citados ou que não estejam de acordo com os critérios do trabalho. Além disso, foram realizadas reuniões quinzenais para discussão e ampliação do conhecimento acerca do estudo proposto. O enfoque da pesquisa será descritivo, o qual explicará a problemática a partir de referenciais teóricos (GIL, 2006).

A partir da leitura e análise do conteúdo, os artigos foram organizados de forma numérica com as principais informações, posteriormente foram criadas duas categorias, sendo elas Categoria 1: Funcionamento CAP X BASE X ACID, Categoria 2: Integração da ACID em Bancos NoSQL, relacionadas aos 5 artigos selecionados. A escolha do assunto aconteceu devido ao interesse em conhecer a integração das características ACID em bancos de dados NoSQL, por meio de revisão da literatura. As buscas das fontes foram realizadas através de leitura concorrente que proporcionam conhecimento científico sobre o tema proposto, periódicos científicos disponíveis através de artigos científicos, artigos disponíveis online e livros.

O trabalho de pesquisa foi realizado nas bases de dados: Sciencedirect, ST Cloud State University, Caeasses. A seleção destes artigos ocorreu através da leitura a fim de selecionar os que estão de acordo com os objetivos desta pesquisa. A composição de cinco momentos específicos é uma ferramenta norteadora aos autores da pesquisa para que os procedimentos sejam estruturalmente descritos de forma coesa na montagem do trabalho. Sendo compostos por: a determinação dos objetivos do trabalho, buscas emanando por seleção e aplicação dos critérios de inclusão, publicações no idioma português e inglês que estiveram em concordância com a pergunta norteadora, que tenham sido publicados no período de 2012 a 2021 e que possuam texto disponível online na íntegra. Os critérios de exclusão são artigos anteriores ao ano de 2012.

Foram realizadas buscas no período de agosto a outubro de 2021 iniciando a busca na Sciencedirect, onde foram disponibilizadas 243 publicações utilizando de forma combinada os termos “NoSQL” e “ACID”, refinando em texto completo, idioma em português e inglês e ano

de publicação entre 2012 e 2020 foram selecionados 200 artigos, destes foram incluídos 2 artigos. Buscando na plataforma ST Cloud State University foram ofertadas 12 publicações combinando os termos “NoSQL” e “ACID”, com aplicação dos filtros e leitura dos textos na íntegra foi incluso 1 nesta pesquisa. As buscas feitas pela base Caesses proporcionou 17 artigos combinando os termos “NoSQL” e “ACID”, com a filtragem e leitura dos textos restaram 1 artigo para a pesquisa. Após a filtragem dos textos procedemos com as leituras exploratória e analítica dos 4 artigos selecionados para esta pesquisa.

Foram realizadas buscas na plataforma digital do MongoDB utilizando os termos “NoSQL” e “ACID” para a busca que resultou em 1 artigo para a pesquisa.

Finalmente foram feitas leituras exaustivas dos artigos, e análise das publicações elencadas para fazer parte do estudo, partindo para a organização dos artigos para beneficiar as características dos 6 artigos selecionados.

3 ANÁLISE E DISCUÇÃO DOS RESULTADOS

3.1 ARTIGO 1. A Critical Comparison of NOSQL Databases in the Context of ACID and Base.

Autor e ano: DEEPAK GC, 2016.

Objetivo: Discutir dois grandes tipos de bancos de dados Relacional e NOSQL e Analisar os diferentes modelos utilizados por essas bases de dados.

Metodologia: Este artigo explora algumas dessas abordagens e explica por que os bancos de dados NOSQL não podem simplesmente seguir o modelo ACID. Quais são as razões por trás do uso extensivo do modelo BASE? Quais são algumas das vantagens e desvantagens de não usar o ACID? Atenção especial será dada para analisar se um modelo é melhor ou superior ao outro. Essas perguntas serão respondidas revendo pesquisas existentes realizadas em algumas das bases de dados NOSQL, com o Cassandra, DynamoDB, MongoDB e Neo4j.

Resultados: Os bancos de dados NOSQL usam o modelo BASE porque não costumam estar em conformidade com o modelo ACID, algo usado por bancos de dados relacionais. No entanto, algumas bases de dados NOSQL adotam abordagens e técnicas adicionais para fazer com que o banco de dados cumpra o modelo ACID.

Link: Disponível em: <https://repository.stcloudstate.edu/msia_etds/8/s>

Acesso em: out.21.

3.2 ARTIGO 2. A middle layer solution to support ACID properties for NoSQL databases.

Autor e ano: LOTFY, *et al.*, 2015.

Objetivo: O principal objetivo deste artigo é manter os pontos fortes dos RDBMSs como propriedades de consistência e ACID e, ao mesmo tempo, fornecer os benefícios que inspiraram o movimento NoSQL através de uma camada média.

Método: Propor um protocolo de compromisso de quatro fases para garantir: o uso de dados recentes, o uso da técnica pessimista para proibir outros que lidam com dados enquanto são usados e as atualizações de dados residentes em muitos locais para evitar a perda de dados.

Resultados: A camada média continua rastreando todas as transações em execução e gerência com outras camadas a execução de transações simultâneas. Esta solução ajudará a aumentar tanto a escalabilidade quanto o throughput. Finalmente, os resultados experimentais mostram que o throughput do sistema melhora o aumento do número de camadas médias em cenários e a quantidade de atualizações para ler em uma transação aumenta.

Link: Disponível em: <<http://dx.doi.org/10.1016/j.jksuci.2015.05.003>>

Acesso em: out.21.

3.3 ARTIGO 3. A proposal of a transaction processing method for MongoDB.

Autor e ano: KUDO, *et al.*, 2016.

Objetivo: Avaliar como os documentos plurais podem ser atualizados como uma transação por este método. Além disso, o nível de isolamento pode ser especificado para cada transação.

Método: Propõem um método para processar dados plurais como uma única transação para o MongoDB, que é uma espécie de banco de dados NoSQL orientado a documentos. Concretamente, cada dado tem os campos de antes e atualização, e o estado da transação é gerenciado.

Resultados: Por este método, mostram que os dados plurais podem ser atualizados como uma transação com o nível de isolamento especificado.

Link: Disponível em: <<https://doi.org/10.1016/j.procs.2016.08.251>>

Acesso em: out.21.

3.4 ARTIGO 4. What are ACID Transactions?

Autor e ano: MONGO DB, 2020.

Objetivo: As transações ACID garantem que um banco de dados estará em um estado consistente após a execução de um grupo de operações.

Método: Neste artigo, os autores exploraram o que é uma transação ACID, como implementar uma transação no MongoDB e por que e quando usá-la.

Resultados: As transações do MongoDB funcionam de forma semelhante às transações em outras bases de dados. Para usar uma transação, inicie uma sessão do MongoDB através de um driver. Em seguida, use essa sessão para executar seu grupo de operações de banco de dados. Você pode executar qualquer uma das operações CRUD (criar, ler, atualizar e excluir) em vários documentos, várias coleções e vários fragmentos.

Link: Disponível em: <<https://www.mongodb.com/basics/acid-transactions>>

Acesso em: out.21.

3.5 ARTIGO 5. SQL vs. NoSQL vs. NewSQL- A Comparative Study.

Autor e ano: BINANI *et al.*, 2016

Objetivo: Discutir sobre SGBDs relacionais e NoSQL e tentar encontrar a solução ideal para requisitos de Big Data.

Método: Com o advento do BigData os SGBDs relacionais não estavam conseguindo processar todos os dados, ele trabalha de maneira estruturada em formato de tabela e relações, e os dados estavam vindo de forma desestruturada e o NoSQL chegou para fornecer a escalabilidade e plataforma não estruturada para aplicativos de Big Data.

Resultados: NoSQL forneceu grandes promessas de ser um banco de dados perfeito para aplicações de Big Data. No entanto, fica aquém por causa de algumas desvantagens importantes, como o NoSQL não garantir as propriedades ACID (atomicidade, consistência, isolamento e durabilidade) de sistemas SQL. Também não é compatível com versões anteriores do banco de dados.

Link: Disponível em: <https://caeaccess.com/archives/volume6/number1/binani-2016-cae-652418.pdf>

Acesso em: out.21

Depois de categorizar e organizar os artigos selecionados, foram criadas duas categorias: Categoria 1 Funcionamento ACID e Categoria 2 Integração ACID em bancos de dados NoSQL.

3.6 CATEGORIA 1: Funcionamento CAP X BASE X ACID

O artigo 1 traz como proposta o teorema da CAP, que surgiu em 2000 pelo Dr. Eric Brewer, que defendia que para se obter um sistema de dados compartilhados em rede deveria-se considerar três componentes: consistência, disponibilidade e tolerância à partição, também conhecidos como CAP. Para um funcionamento eficiente e ainda para um resultado satisfatório, defendia a necessidade do uso de pelo menos dois dos três componentes aplicando-os nos bancos de dados NoSQL (DEPAC, 2016).

Os artigos 1, 2 e 5 abordam as propriedades BASE, que advém do Teorema de CAP. BASE traz uma proposta de modelo optativo, ACID que diferencia do Teorema de CAP limitado a usar pelo menos dois dos três componentes, BASE demonstra que pode deixar os dados em um estado mais consistente em uma operação dispensando a consistência durante todo o tempo. Diferentemente da ACID, onde qualquer transação precisa ser consistente, nenhuma outra operação pode ser feita enquanto uma está sendo executada (DEPAC, 2016, LOTFY, et al., 2015, BINANI et al., 2016).

Atomicidade, consistência, isolamento e durabilidade (ACID) é o conjunto de propriedades de transação em banco de dados. As propriedades ACID são abordadas nos artigos 1, 2, 4 e 5 apresentando e os SGBDS relacionais que contém essa característica ACID é possível permitir uma maior consistência de dados na ocorrência de transações, implementados para melhora na garantia das operações, fazendo com que a solicitação feita seja realmente efetuada, mesmo quando ocorra mais de uma operação ao mesmo tempo. Quando uma operação se inicia a outra termina, finalizando e garantindo que as operações feitas realmente sejam salvas e mesmo se ocorram falhas após as operações, os dados permanecerão íntegros e disponíveis para as próximas consultas ou alterações (DEPAC, 2016, LOTFY, et al., 2015, MONGO DB, 2020, BINANI et al., 2016).

3.7 CATEGORIA 2: Integração da ACID em Bancos NoSQL

O surgimento do BIG DATA acarretou muitas aplicações web que começaram a sofrer com os bancos de dados relacionais, são muitos dados que são armazenados e consultados e fazer isso com um banco estruturado estava ficando inviável, a quantidade de dados era gigantesca para o tempo de pesquisa, com isso veio os bancos não estruturados conhecemos

como NoSQL, ele também deixou de lado a linguagem estruturada que era bastante conhecida como SQL, um banco com escalonamento horizontal, não estruturado, não dependendo de SQL e ainda fornecia 4 modelos de armazenamento podendo escolher o que melhor lhe atendia, os artigos 1, 2, 5, trata-se desse surgimento desse novo modelo de banco de dados (DEPAC, 2016, LOTFY, et al., 2015, BINANI et al., 2016).

Como toda ferramenta que surge vem também novos problemas, diferente dos SGBDS relacionais os bancos NoSQL tem um problema com integridade dos dados, o teorema CAP e a propriedade BASE não conseguem dar a confiabilidade que as propriedades ACID oferece no modelo relacional, por mais que o NoSQL armazena dados big data seja infinitamente melhor que os relacionais o fato de que o estado final do banco após inserções ou consultas não serem as mesmas se pensou no fato de implementarem as características ACID nos bancos NoSQL ter toda escalabilidade que ele oferece e ainda tem essas características que vai melhorar o estado inicial e final de cada transação (DEPAC, 2016, LOTFY, et al., 2015, BINANI et al., 2016).

Existem duas formas de armazenar dados que dão certo e cada uma se propõe com uma finalidade diferente. Uma com dados estruturados e outra com dados escalados. O Artigo 5 explica que o NewSQL é uma tentativa de unir esses dois modelos, o modelo relacional oferece dimensionamento vertical, O NewSQL oferece dimensionamento vertical e horizontal sendo distribuídos e ainda oferecendo as propriedades ACID, são excelentes para trabalhar com cargas muito grandes de dados. O NewSQL é um banco relacional combinado com o NoSQL que suporta nuvem, o ponto negativo que a tem muita complexidade de consulta muito alta (BINANI et al., 2016).

O artigo 3 propõe a ideia do uso de isolamento específico para cada transação pois, antes a transação era feita uma consulta nos dados e após toda a transação os dados eram consultados novamente. A ideia é que tenha sempre dois estados do banco, o primeiro antes de ocorrer a alteração e outro depois de ocorrer a transação, essas ideias surgem pelo fato de que essas atualizações podem ser consultadas por outras transações que as propriedades ACID não poderiam manter. Utilizando esse método de isolamento específico garante que apenas aqueles dados serão atualizados porem as outras características da ACID se perdem (KUDO, et al., 2016).

4 CONSIDERAÇÕES FINAIS

Dessa forma, conclui-se que ACID nos SGBDs relacionais tem um papel importante, pois essas características ajudam o banco ser íntegro, caso haja algum erro, seja queda de energia na inserção ou atualização de um dado, por exemplo, o banco vai voltar ao último estado válido isso e a atomicidade. A consistência age para sempre deixar o banco em um estado válido, que nunca vai estar com uma informação errada devido a algum erro de sistema ou informações que são concorrentes elas serão isoladas. Caso tenha uma mesma informação sendo acessada ao mesmo tempo e ocorra uma alteração, por ambas as partes, o sistema vai validar o que fez primeiro, dando um erro para o posterior, a durabilidade vai garantir que após as inserções, alterações o banco vai estar no último estado que foi deixado, validando todas as informações o conjunto dessas características fizeram os bancos NoSQL olharem para elas devido aos problemas que acontecia de integridade ou demora na consulta dos dados, começaram a se importar com a consistência do banco, sempre deixar ele válido, adaptando o que a ACID proporciona para o sistema NoSQL, continua com a mesma base, apenas alterando como os dados são tratados dando mais segurança e disponibilidade para o banco de dados.

REFERÊNCIAS BIBLIOGRÁFICAS

ALECRIM, Emerson. **O que é Big Data?**, 2015. Disponível em: <https://www.infowest.com/big-data.php>. Acesso em 22/03/2021.

AWS, **Grafos em bancos de dados**. Disponível em: <https://aws.amazon.com/pt/nosql/graph/> Acesso em: 14/04/2021.

BARROSO, Isaias; **Banco de dados, 2012**. Disponível em: <https://isaiasbarroso.wordpress.com/2012/06/20/banco-de-dados-orientado-a-colunas/> Acesso em 28/04/2021.

BREWER, Eric. **Towards robust distributed systems**. 2000. Disponível em: https://sites.cs.ucsb.edu/~rich/class/cs293b-cloud/papers/Brewer_podc_keynote_2000.pdf. Acesso em 29/04/2021

BRITOS, Ricardo; **Bancos de Dados NoSQL x SGBDs Relacionais: Análise Comparativa**, 2010. Disponível em: <http://docplayer.com.br/433629-Bancos-de-dados-nosql-x-sgbds-relacionais-analise-comparativa.html>. Acesso em: 01/11/2021.

CIFERRI, Cristina Dutra de Aguiar. **Banco de Dados NoSQL: Processamento Analítico de Dados**. Disponível em: <http://wiki.icmc.usp.br/images/1/18/SCC0542012017noSQL.pdf> Acesso em: 28/04/2021.

CODD, Edgar Frank. **A Relational Model of Data for Large Shared Data Banks**, 1970. Disponível em: <https://www.seas.upenn.edu/~zives/03f/cis550/codd.pdf> Acesso em 02/11/2020.

COUCHBASE, **Introduction to Couchbase**. Disponível em: <https://docs.couchbase.com/couchbase-devguide-2.0/> Acesso em: 28/04/2021.

ELMASRI, Ramez; NAVATHE, Shamkant B, **SISTEMAS DE BANCO DE DADOS**. São Paulo: Pearson Education 2005. Acesso em 05/11/21.

EQUIPO DE EXPERTOS, **Lenguaje SQL, história y conceptos básicos, In: Universidad Internacional de Valencia, Valencia (Espanha)**. Disponível em: <https://www.universidadviu.com/co/actualidad/nuestros-expertos/lenguaje-sql-historia-y-conceptos-basicos> Acesso em 05/05/2021.

FOWLER, Martin. **NoSQL Definition**. Martin Fowler, 2012. Disponível em: <https://martinfowler.com/bliki/NosqlDefinition.html>. Acesso em: 14/09/2020.

GIL, A.C. **Como elaborar Projetos de Pesquisa**. 4.ed. São Paulo: Atlas, 2006.

GUTIERRY, Antonio Neto. Conheça a geração de banco de dados NoSQL e NewSQL. **Devmedia**, 2015. Disponível em: <https://www.devmedia.com.br/conheca-a-geracao-de-banco-de-dados-nosql-e-newsq/33202>. Acesso em: 26/04/2021.

KOO, Lawrence. **RESENHA DO LIVRO BIG DATA**, 2014. Disponível em: <https://www.redalyc.org/jatsRepo/3502/350260805010/html/index.html#:~:text=RESENHA%20DO%20LIVRO%20BIG%20DATA&text=O%20autor%20C%3%A9zar%20Taurion%20relata,dos%20movimentos%20da%20rede%20social>. Acesso em: 17/10/2020.

LEMOS, P H. S; FIGUEIREDO, P. S. **Uma Análise dos Novos Sistemas de Bancos de Dados Relacionais Escaláveis**, 2014. Disponível em: <http://repositorio.poli.ufrj.br/monografias/monopoli10010084.pdf> Acesso em: 17/05/2021.

MongoDB. **Notas de versão para MongoDB 4.0**, 2018. Disponível em: <https://docs.mongodb.com/v4.0/release-notes/4.0/> Acesso em: 20/11/2020.

MICREIROS.COM, **Tipos de bancos de dados NoSQL** Disponível em: <https://micreiros.com/tipos-de-bancos-de-dados-nosql/>. Acesso em: 26/04/2021.

NoSQL-Database.org, 2009. Disponível em: <https://hostingdata.co.uk/nosql-database-org-joins-hostingdata-co-uk/> Acesso em: 22/04/2021.

PAVLO, A; ASLLET, M. **What 's Really New with NewSQL?**, 2016. Disponível em: <https://db.cs.cmu.edu/papers/2016/pavlo-newsql-sigmodrec2016.pdf> Acesso em: 22/04/2021.

PEREIRA, Altieri. **SQL Server: ACID (Atomicidade, Consistência, Isolamento e Disponibilidade)**, 2015. Disponível em: <https://altieripereira.wordpress.com/2015/07/16/sql-server-acidatomicidade-consistencia-isolamento-e-disponibilidade/> Acesso em: 26/04/2021.

REIS, Fábio. **Conceito de Banco de Dados: O que significa ACID**. 2018. Disponível em: <http://www.bosontreinamentos.com.br/bancos-de-dados/conceitos-de-bancos-de-dados-o-que-significa-acid/> Acesso em: 26/04/2021.

SANTOS, Lucas. **O teorema de CAP – Utilizado para auxiliar um servidor de sistema distribuídos**. 2017. Disponível em: <https://medium.com/@lucascodejs/o-teorema-de-cap-utilizado-para-auxiliar-um-servidor-de-sistema-distribuidos-f73d6a4c5d9>. Acesso em: 12/06/2021.

SILBERSCHATZ, A; KORTH, H. F; SUDARSHAN, S. **Sistema de Banco de Dados**. 7ª Edição. Rio de Janeiro, Editora GEN LTC, 2020. Acesso em: 26/04/2021.

SILVA, Alexandre, **O Fantástico Mundo do NoSQL!**, 2017. Disponível em: <https://medium.com/@albsilva/o-fantastico-mundo-do-nosql-2e72c5640e69> Acesso em 06/05/2021.

SIMON, Imre. **A revolução digital e a sociedade do conhecimento: o que é informação? como ela age?**, 1999. Disponível em: <http://www.ime.usp.br/~is/ddt/mac333/aulas/tema-11-24mai99.html>. Acesso em: 18/11/2021.

STROZZI, Carlo. **NoSQL: a non-SQL RDBMS**, 1998. Disponível em: http://www.strozzi.it/cgi-bin/CSA/tw7/I/en_US/NoSQL/Home%20Page Acesso em: 26/04/2021.

Thibault, Dory, **Quand et pourquoi utiliser une base de données NoSQL?**
Disponível em: <https://www.marginweb.com/blog/quand-et-pourquoi-utiliser-une-base-de-donnees-nosql>.
Acesso em 06/05/2021.

VALE, Vinicius. **NoSQL**. 2015. Disponível em: <http://blog.sudoers.com.br/nosql/>. Acesso em: 26/04/2021.